## **vn.py快速入门8 - 策略实盘自动交易**

原创 用Python的交易员 VNPY  2019-12-03

收录于话题

#vn.py快速入门

8个

跑完了历史数据回测和优化，得到了一个不错的回测资金曲线，最后就可以准备开始实盘交易了。在教程2-5中我们已经接触了真实账户和仿真账户的概念，这里强调一个原则：

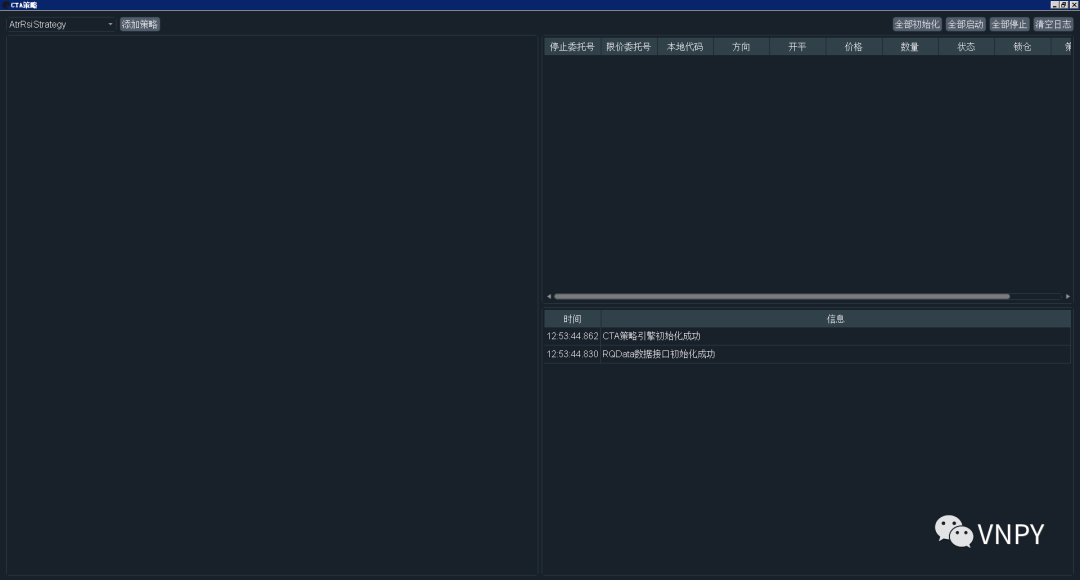
所有量化策略在开始真金白银交易之前，都应该经过仿真账户的充分测试，毕竟每个人交易的本金都来之不易，一定要有十分负责的态度。

本篇教程我们继续以股指期货为例，其他产品的量化策略实盘也基本都一样。首先启动VN Trader Pro，加载CTP接口以及CTA策略模块（CtaStrategy），或者也可以直接运行VN Trader Lite。

进入VN Trader主界面后，连接登录CTP接口，等待日志组件中看到“合约信息查询成功”的信息输出。

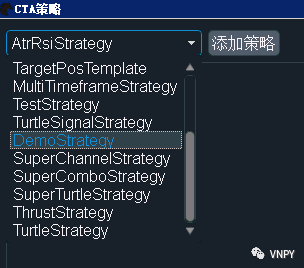
**加载一个实例**

随后，点击菜单栏的“功能”->“CTA策略”，或者左侧导航栏里的国际象棋棋子的图标，看到CTA策略实盘交易的窗口：

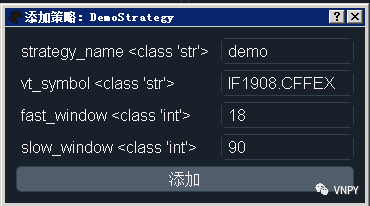


此时在右下方的日志监控组件中，可以看到“RQData数据接口初始化成功”的信息，如果没有的话请照着上一篇教程里的方法，配置RQData数据服务。

接下来要基于之前开发好的策略模板（类），来添加策略的实例（对象），点击右上角的策略下拉框，找到DemoStrategy：



点击添加策略按钮，出现策略实盘参数配置的对话框：



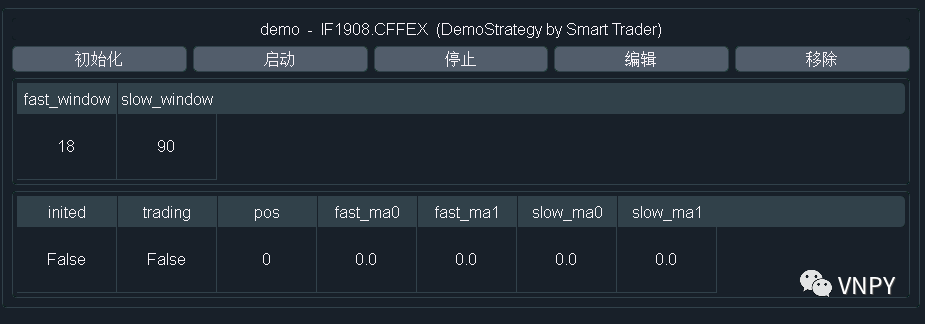
每个参数字段，后面的<>括号中说明了该字段对应的数据类型，注意必须根据要求填写，否则实例创建会出错。

首先我们要给策略实例一个名字，也就是**strategy\_name**字段，注意每个实例的名称必须唯一，不能重复。

然后需要指定策略具体要交易的合约，通过本地代码**vt\_symbol**来指定（合约代码 + 交易所名称）。注意在上一篇教程中我们回测使用的是RQData提供的股指连续合约数据IF88.CFFEX，该合约只是为了方便回测而人工合成的数据，在实盘交易系统中并不存在。在实盘中，需要指定具体的交易合约，一般选择该期货品种当前流动性最好的月份，比如写本文时是2019年8月3日，此时的股指主力合约为IF1908.CFFEX。

fast\_window和slow\_window是策略里写在parameters列表中的参数名，这里我们填入上一篇教程中优化后的结果18和90。

点击“添加”按钮后，在左侧的策略监控组件中，就能看到该策略实例了：



顶部按钮用于控制和管理策略实例，第一行表格显示了策略内部的参数信息，第二行表格则显示了策略运行过程中的变量信息（变量名需要写在策略的variables列表中）。**inited**字段表示当前策略的初始化状态（是否已经完成了历史数据回放），**trading**字段表示策略当前是否开始交易。

注意上方显示的所有变量信息，需要在策略中**调用put\_event函数**，界面上才会**进行数据刷新**。有时用户会发现自己写的策略，无论跑多久这些变量信息都不动，这种情况请检查策略中是否漏掉了对put\_event函数的调用。

**策略初始化**

点击监控组件顶部的“初始化”按钮，此时内部的CTA策略引擎会先调用策略的on\_init函数，运行用户定义的逻辑，随后按照顺序完成以下三步任务。

**获取历史数据**

首先是载入该合约最新的历史数据，具体载入数据的长度，通过策略内部的load\_bar函数的参数控制。数据载入后会以逐根K线（或者Tick）的方式推送给策略，实现内部变量的初始化计算，比如缓存K线序列、计算技术指标等。

在载入时，CTA策略引擎会**优先通过RQData**来获取历史数据，RQData的数据服务提供盘中K线更新，因此即使在9点45分才启动策略，也能获取到之前从9点30开盘到9点45分之间的K线数据提供给策略进行初始化计算，而不用担心数据缺失的问题。

如果RQData不支持该合约（比如数字货币、外盘期货），CTA策略引擎则会**尝试使用交易接口**进行获取。对于所有的数字货币接口以及IB接口来说，交易的服务端系统都提供了相应的历史数据下载功能。

最后，如果交易接口也获取不到，那么CTA策略引擎会**访问本地数据库**来加载历史数据。这种情况下，用户需要自己来保证数据库中的数据完整性（满足需求），比如通过DataRecorder录制，使用CsvLoader从CSV文件载入，或者使用其他数据服务（比如万得宏汇等）来更新。

**载入缓存变量**

量化策略在每天实盘运行的过程中，有些变量纯粹只和行情数据相关的，这类变量通过上一步的加载历史数据回放就能得到正确的数值。另一类变量则可能是和交易状态相关的，如策略的持仓、移动止损的最高价跟踪等，这类变量需要缓存在硬盘上（退出程序时），第二天回放完历史数据后再读取还原，才能保证和之前交易状态的一致性。

在CtaStrategy中这一步骤对于用户来说是**几乎无感的**，每次关闭程序时会自动将每个策略的variables列表对应的变量写入json文件（缓存在.vntrader目录下的**cta\_strategy\_data.json**中），并在下一次策略初始化时自动载入。

注意在某些情况下，可能缓存的数据出现了偏差（比如手动平仓了），此时可以通过**手动修改json文件来调整**。

**订阅行情推送**

最后是获取该策略所交易合约的信息（基于vt\_symbol），并订阅该合约的实时行情推送，如果找不到该合约的信息，比如没有登录接口或者vt\_symbol写错了，则会在日志模块中输出相应的报错信息。

注意对于IB接口来说，因为登录时无法自动获取所有的合约信息，只有在用户手动订阅行情时才能获取，因此**需要在主界面上先行手动订阅该合约行情**，然后再点击“初始化”按钮。



以上三个步骤全部完成后，策略的inited状态变为True，且变量也都有了对应的数值（不再为0），则说明初始化已经完成。

**策略的启动**

完成策略初始化后（inited状态为True时），才可以点击“启动”按钮启动策略的自动交易功能：



当trading状态为True时，策略内部的交易请求类函数（buy/sell/short/cover/cancel\_order等），以及信息输出类函数（write\_log/send\_email等），才会真正执行并发出对应的请求指令到底层接口中（真正执行交易）。

在上一步策略初始化的过程中，尽管策略同样在接收（历史）数据，并调用对应的功能行数，但因为trading状态为False，所以并不会有任何真正的委托下单操作或者日志信息输出。

**策略的停止**

到了市场收盘时间，或者盘中遇到紧急情况时，点击“停止”按钮即可停止策略的自动交易。

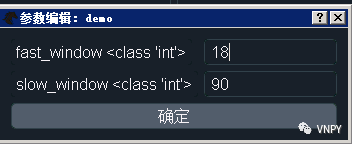
CTA策略引擎会自动将该策略之前发出的所有活动委托全部撤销（保证在策略停止后不会有失去控制的委托存在），同时执行上面提到过的变量缓存操作。

这两步都完成后，策略的trading状态会变为False，此时可以放心的关闭程序了。

在CTA策略的实盘交易过程中，正常情况应该让策略在整个交易时段中都自动运行，尽量不要有额外的暂停重启类操作。对于国内期货市场来说，应该在夜盘时段开始前，启动策略的自动交易，然后直到第二天下午收盘后，再关闭自动交易，中间的夜盘收盘属于同一交易日内，无需停止策略。

**编辑和移除**

跑量化策略的过程中，有时可能需要调整策略的参数，点击策略监控组件上的**“编辑”**按钮，即可在弹出的参数编辑对话框中任意修改参数：



点击确定按钮后相应的修改会立即更新在参数表格中，注意策略实例的交易合约代码无法修改，同时修改完后也不会重新执行初始化操作。

为了安全起见，请一定要在trading状态为False时（自动交易停止），才进行参数的编辑操作。

想要删除某个策略实例时，点击**“移除”**按钮，CTA策略引擎会自动完成该策略实例的对象销毁和内存释放，并在GUI图形界面上移除其监控组件，此时该策略的名称（strategy\_name）也可以再次使用。注意只能移除trading状态为False的策略实例，如果策略已经启动了自动交易功能需要先停止。

**其他**

在每天的实盘交易中，如果存在比较多的策略实例，可以通过右上角的**“全部初始化”**、**“全部启动”**和**“全部停止”**三个按钮，来一次性对所有的策略实例进行相应的操作管理，避免开盘前点几十下“启动”，收盘后点几十下“停止”的重复劳动。

当日志监控组件中的信息条数过多时，可以点击右上角的**“清空日志”**按钮来清空其中已有的信息。

最后还剩没提到的就是右上方区域的停止单（Stop Order）监控组件，该组件用来跟踪所有CTA引擎内的本地停止单状态变化，具体用法请参考官网文档：<https://www.vnpy.com/docs/cn/index.html>。

实盘交易中除了每天机械的启动和关闭策略外，更重要的是对每天策略运行交易结果的跟踪和总结：逻辑运行和回测是否一致、实盘成交和回测假设的滑点偏差有多少等等，并用这些观察到的结果数据，去修正回测研究中用到的参数假设，从而才能实现自己策略研究水平的迭代进步。